Decoupled Representation Learning for Skeleton-Based Gesture Recognition

Jianbo Liu[†‡] Yongcheng Liu[†‡] Ying Wang[*†] Véronique Prinet[†] Shiming Xiang[†‡] Chunhong Pan[†]

[†]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[‡]School of Artificial Intelligence, University of Chinese Academy of Sciences
Email: {jianbo.liu, yongcheng.liu, ywang, prinet, smxiang, chpan}@nlpr.ia.ac.cn

Abstract

Skeleton-based gesture recognition is very challenging, as the high-level information in gesture is expressed by a sequence of complexly composite motions. Previous works often learn all the motions with a single model. In this paper, we propose to decouple the gesture into ***hand posture variations*** *and* ***hand movements****, which are then modeled separately. For the former, the skeleton sequence is embedded into a 3D hand posture evolution volume (****HPEV****) to represent fine-grained posture variations. For the latter, the shifts of hand center and fingertips are arranged as a 2D hand movement map (****HMM****) to capture holistic movements. To learn from the two inhomogeneous representations for gesture recognition, we propose an end-to-end two-stream network. The HPEV stream integrates both spatial layout and temporal evolution information of hand postures by a dedicated 3D CNN, while the HMM stream develops an efficient 2D CNN to extract hand movement features. Eventually, the predictions of the two streams are aggregated with high efficiency. Extensive experiments on SHREC'17 Track, DHG-14/28 and FPHA datasets demonstrate that our method is competitive with the state-of-the-art.*

1. Introduction

Gesture recognition is an attractive research topic due to its wide range of applications in many fields, *e.g.* assisted living, virtual game control and sign language interpretation. Early works for this task are mainly based on RGB videos or depth sequences. However, both modalities have the drawback of plausible background changes, which is often harmful to gesture recognition. More recently, hand skeleton has become a popular modality. Skeleton inherently highlights the key information of hand gestures, whilst being robust to various illuminations and complex backgrounds. Accordingly, skeleton-based gesture recognition has drawn much attention in recent years.

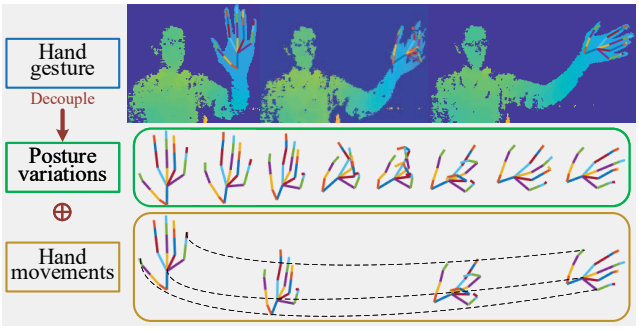

Figure 1. The motivation of this paper. Dynamic hand gesture can be decomposed into hand posture variations and hand movements.

Motivated by the remarkable success of deep learning in vision tasks, much effort has focused on applying CNN [9, 19, 23, 25], RNN [10, 34] and LSTM [22, 25, 30] to skeleton-based gesture recognition. In these methods, hand skeletons are usually constructed as a sequence of joint-coordinate vectors or a pseudo image. More recently, some authors apply graph convolutional network (GCN) to analyze hand skeletons [4]. They often embed skeleton sequences into a predefined spatio-temporal graph structure. These deep learning-based methods represent skeleton sequences in a predefined and fixed structure. Hand gestures always contain the interactions of different joints. Aggregating the local features of these joints is crucial for hand gesture recognition. However, the fixed structure for skeletons could be under-effective to capture local features for all interactive joints, since these joints may be nonadjacent in the predefined structure. Moreover, most of these methods work with a single deep model, which may have difficulty in analyzing the complexly composite hand gestures.

As is shown in Figure 1, a dynamic hand gesture can be decomposed into hand posture variations and hand movements. Previous deep learning based works aim to learn the

*Corresponding author: Ying Wang

2575-7075/20/$31.00 ©2020 IEEE
DOI 10.1109/CVPR42600.2020.00579

two features in a single stream. However, using a single network to learn these two features may yield sub-optimal performance. Motivated by this observation, we propose to decouple the dynamic gesture into *hand posture variations* and *hand movements*. The former captures the spatial layout changes of hand joints, while the latter reflects global motion trajectories. Intuitively, the great difference between the two representations motivates us to develop a two-stream framework for gesture recognition, which is illustrated in Figure 2.

Specifically, two representations for hand posture variations and hand movements are first decoupled from raw skeleton sequence and then learned by two networks separately. For posture variations, skeleton sequence is embedded into a Hand Posture Evolution Volume (HPEV) with a gap between adjacent skeletons, ignoring hand relative positions. In this way, the spatial and temporal information of posture evolution can be encoded effectively. Then a 3D CNN based network (named, HPEV-Net) is designed to aggregate both the spatial layout and temporal evolution information. By volume representation and using 3D CNN based network, we can alleviate the limitations imposed by the above mentioned predefined-structure methods. Since the interactive joints tend to be adjacent to each other in volume space, the 3D convolution kernel can aggregate the local features of these joints naturally. Nevertheless, the volume representation may lose details of subtle hand motion due to the resolution restriction. As a remedy, we introduce Fingertip Relative Position Vector (FRPV) as a compensatory cue for subtle gesture recognition. For hand movements, the shifts of both fingertips and hand center are arranged as Hand Movement Map (HMM). Then, a 2D CNN based network (named, HMM-Net) is applied to learn the hand movement features. Eventually, recognition results can be obtained by fusing the predictions of the two networks.

In summary, the main contributions of this paper can be summarized as follows:

- We propose a novel method for gesture modeling: we represent the gesture as hand posture variations and hand movements and model them separately by Hand Posture Evolution Volume (HPEV) and Hand Movement Map (HMM). The method can simultaneously enhance the expressive power of the posture and motion information.
- We introduce a unified and efficient two-stream framework to effectively learn the decoupled representations. Extensive experiments on benchmarks demonstrate its superior performance in gesture recognition.

2. Related work

With the success of deep learning, many works applied deep learning to gesture and action recognition in an end-to-end manner. By transforming the raw skeleton sequence to a pseudo image, CNN is used to extract gesture and action features [9, 19, 23, 25]. RNN [10, 34] and LSTM [22, 25, 30] are wildly used to capture spatial and temporal relationships among joints. Moreover, attention mechanism based approaches are applied in [35]. Recently, deep learning on manifolds and graphs has increasingly attracted attention. The goal is to improve gesture and action recognition using manifold learning [24] and GCN [36]. Most of these methods model skeleton sequences in a predefined and fixed structure. Therefore, they cannot capture the local features of interactive joints efficiently. This drawback is alleviated by spatial and temporal volume modeling for skeletons [32]. However, skeletons are embedded in the volume corresponding to the whole motion region. Gestures with significant movements may suffer from low-resolution representation for hand skeletons leading to inferior performance. In this paper, we ignore hand relative positions when building Hand Posture Evolution Volume (HPEV). Hence, the skeletons can be represented with high resolution.

Liu *et al*. [21] first proposed a two-stream network for action recognition. They introduce shape and motion evolution maps to represent action and use the two-stream network to learn these features. Compared with this method, our approach models hand posture variations and hand movements using two different modalities, which is more effective. Moreover, due to the different skeletal topologies, actions and gestures have different motion characteristics. The representation dedicated to action recognition may not be optimal for gesture recognition. Compared to actions, gestures can be clearly decomposed into two parts: the hand posture changes and the hand movements. Precisely inspired by this observation, we design a specialized decoupled representation to improve gesture recognition.

3. Proposed method

The overall architecture of the proposed method is shown in Figure 2. Given a skeleton sequence, a hand posture evolution volume (HPEV) and a hand movement map (HMM) are generated to represent hand posture variations and hand movements respectively. We design a 3D CNN based HPEV-Net for learning a discriminative feature vector of hand posture variations from HPEV. As a supplementary clue, a fingertip relative position vector (FRPV) is concatenated to this feature vector. Simultaneously, the HMM is fed into a CNN based HMM-Net to capture hand movement information. Each sub-network is terminated by a fully connected layer. Finally, the classification scores of two subnets are fused to produce the final prediction.

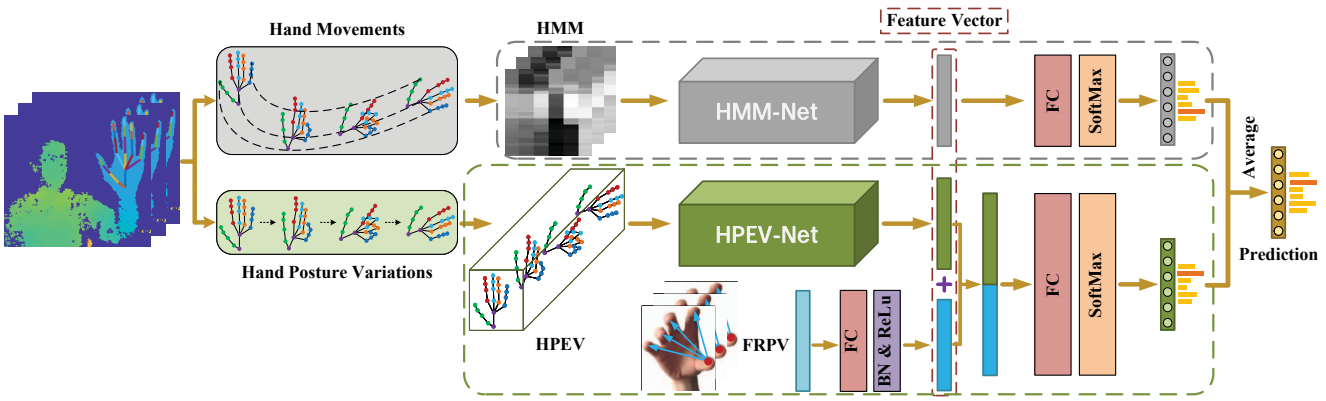

Figure 2. The framework architecture of our two-stream network. The gesture is decoupled into the hand posture evolution volume (HPEV) and the hand movement map (HMM) to represent hand posture variations and hand movements. The HPEV-Net and HMM-Net extract two feature vectors for these characteristics respectively. The feature of the fingertip relative position vector (FRPV) is concatenated to the hand posture variations feature vector as a compensate cue for subtle gestures. The scores obtained by the two subnets are averaged to produce the final prediction for inference.

3.1. Hand Posture Volume

In the skeleton-based hand gesture recognition field, hand gesture is described as a hand skeleton sequence. For the n-th hand gesture sample G_n, it can be represented as $G_n = \{S_t^n | t = 1, 2, \ldots, T_n\}$, where S_t^n is the hand skeleton of t-th frame, and T_n is the length of the skeleton sequence. Hand skeleton is a collection of 3D positions of hand joints, *i.e.* $S_t^n = \{\mathbf{p}_{i,t}^n | \mathbf{p}_{i,t}^n = (x_{i,t}^n, y_{i,t}^n, z_{i,t}^n), i = 1, 2, \ldots, J\}$, where $\mathbf{p}_{i,t}^n$ is the 3D coordinates of hand joint i at frame t, and J is the number of hand joints.

Since each skeleton sequence sample has a different length, T_n varies in a large range. In order to fix the input size, it is necessary to process the original skeleton sequences to obtain a constant length T. Specifically, when $T_n > T$, the skeleton sequence is sampled uniformly. For $T_n < T$, some frames are repeated until the skeleton sequence reaches T frames. The sampling processing from G_n to G_n^T with T frames can be formulated as:

$$G_n^T = \left\{ S_\tau^n \middle| \tau = \left\lceil \frac{T_n}{T} \times t \right\rceil, t = 1, 2, \ldots, T \right\}. \quad (1)$$

Before diving into hand posture evolution volume, we first focus on how to construct hand posture volume for encoding the spatial configurations of the hand skeleton. To model the skeleton into a cube volume, the raw coordinates of skeletons should be transformed into the volume coordinates. To this end, the raw coordinates are normalized to $[-1, 1]$, then the normalized coordinates are scaled to the volume coordinates. During hand posture evolution volume modeling, we only focus on the hand posture and its variation, ignoring the hand movements. Thus, when building hand posture volume, each skeleton in the sequence is supposed to be put in the center of volume without considering the hand position. In addition, the skeleton should occupy the volume as much as possible in order to take full advantage of the volume space to model the skeleton with a high resolution. Therefore, we propose to normalize raw coordinates of each skeleton to $[-1, 1]$ with the origin at the center of the volume using the maximum bounding box of skeletons in all training data. More specifically, for the n-th hand gesture G_n, the hand skeleton at frame t is S_t^n. The side lengths of bounding box for the skeleton S_t^n are defined as Δx_t^n, Δy_t^n and Δz_t^n, and they can be formulated as follows:

$$\begin{cases} \Delta x_t^n = \max(x_{i,t}^n) - \min(x_{i,t}^n) \\ \Delta y_t^n = \max(y_{i,t}^n) - \min(y_{i,t}^n) \\ \Delta z_t^n = \max(z_{i,t}^n) - \min(z_{i,t}^n) \end{cases} \quad i = 1, 2, \ldots, J. \quad (2)$$

Subsequently, the maximum side lengths of bounding box for all skeletons are defined as Δx_{max}, Δy_{max} and Δz_{max}, Δx_{max} can be formulated as:

$$\Delta x_{max} = \max(\Delta x_t^n), \quad (3)$$

where $t = 1, 2, \ldots, T$, $n = 1, 2, \ldots, N$ and N is the number of skeleton sequence samples in training data. Δy_{max} and Δz_{max} are obtained in the same way as Δx_{max}. Finally, the raw coordinates of each skeleton are normalized to $[-1, 1]$ using Δx_{max}, Δy_{max} and Δz_{max} as the scale factor for each dimension. Hence, the x-coordinate of each joint for skeleton is normalized as:

$$x_{norm} = \frac{x - \frac{x_{min} + x_{max}}{2}}{\Delta x_{max}} \times 2, \quad (4)$$

where x_{min} and x_{max} are the minimum and maximum x-coordinate values of this skeleton, x is the original x-coordinate, and x_{norm} is the normalized x-coordinate. The

normalization processes of y-coordinate and z-coordinate are implemented in the same way as x-coordinate. By normalization, the center of skeleton is aligned to $(0, 0, 0)$. Next, the normalized coordinates are scaled and discretized to the volume coordinates. If the skeleton is embedded into a cube volume with $R \times R \times R$ resolution, the normalized x-coordinates can be transformed to the volume coordinates according to Eq. (5), where x_{norm} is the normalized x-coordinate and x_v is the x-coordinate under the volume coordinate system, $x_v \in \{1, 2, \ldots, R\}$.

$$x_v = round\left((x_{norm} + 1) \times \frac{R}{2}\right), \tag{5}$$

During hand posture volume modeling, the skeleton is put into a volume space. The values of the occupied voxels equal to 1 and the values of the rest of the voxels equal to 0. In this way, all joints of the skeleton are embedded in the volume, the spatial configurations of joints are encoded naturally. For hand gesture $G_v = \{S_{v,t} | t = 1, 2, \ldots, T\}$, G_v and $S_{v,t}$ indicate that the coordinates of skeletons are transformed into volume coordinates, the hand posture volume for skeleton $S_{v,t}$ can be represented as a tensor V of (R, R, R) dimension, the entry with index (i, j, k) of tensor V can be written as Eq. (6), where $i, j, k = 1, 2, \ldots, R$.

$$V(i, j, k) = \begin{cases} 1, & \text{if } (i, j, k) \in S_{v,t} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

3.2. Hand Posture Evolution Volume (HPEV)

The Hand Posture Evolution Volume (HPEV) accounts for the temporal variations of skeletons. We concatenate all skeletons of a sequence into a volume with a gap between adjacent skeletons, therefore turning the skeleton sequence into a set of point clouds in the volume space. More specifically, with respect to hand gesture G_v, the hand posture volume V_t for $S_{v,t}$ is constructed according to Eq. (6). The final hand posture evolution volume V_{HPEV} for G_v with $(R + \theta(T - 1), R, R)$ dimension can be derived as:

$$V_{HPEV}(i + \theta(t - 1), j, k) = \begin{cases} 1, & \text{if } (i, j, k) \in S_{v,t}, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where $t = 1, 2, \ldots, T$, and θ is the gap between adjacent skeletons. The HPEV encodes the spatial configurations of each skeleton and the posture variations of the gesture.

3.3. Hand Movement Map (HMM)

Hand movements always include the motion of the whole hand and the shift of each joint. For the motion of the whole hand, we use the central point of all hand joints to represent its position. For the shift of joints, five fingertips instead of all joints are used to stand for joint motion. The reason is that when performing hand gestures, the motion of the fingertip is similar to the motion of the other three joints in one finger. Besides, compared with these three joints, the fingertip is always visible and the motion of fingertip is notable. Concretely, for hand gesture $G = \{S_t | t = 1, 2, \ldots, T\}$, the centroid of skeleton $S_t = \{\mathbf{p}_{i,t} | \mathbf{p}_{i,t} = (x_{i,t}, y_{i,t}, z_{i,t}), i = 1, 2, \ldots, J\}$ is:

$$C_t = \frac{1}{J} \sum_{i=1}^{J} \mathbf{p}_{i,t}. \tag{8}$$

Hence, the movements of hand can be formulated as:

$$M_H = \{C_t - C_1 | t = 1, 2, \ldots, T\}. \tag{9}$$

Correspondingly, the movements of fingertips are:

$$M_{F,j} = \{\mathbf{p}_{j,t} - \mathbf{p}_{j,1} | t = 1, 2, \ldots, T\}, \tag{10}$$

where j is the index of the five fingertips.

Finally, we arrange the hand movements and fingertip motion as a Hand Movement Map (HMM), where the frame is treated as the column of the map, five fingertips and central point are treated as the row, the coordinates are treated as the three channels of the map.

3.4. Fingertip Relative Position Vector (FRPV)

Using volume to model the hand skeleton, the spatial configurations of joints are displayed naturally. However, due to the restriction of the resolution, the volume representation is difficult to express subtle finger motion. This is why we introduce the fingertip relative position vector (FRPV), describing finger motion precisely. The shifting of the fingertip is correlated with the other three joints in one finger. Besides, hand gestures always contain the interactions of thumb with four fingers. Therefore, the relative positions of four fingers and thumb of each frame are used to construct the FRPV. In particular, for frame t, the four relative positions are concatenated as a vector $\mathbf{v_t}$ as follows:

$$\begin{aligned} \mathbf{v_t} =& (\mathbf{p}_{I,t}, \mathbf{p}_{M,t}, \mathbf{p}_{R,t}, \mathbf{p}_{L,t}) \\ & - (\mathbf{p}_{0,t}, \mathbf{p}_{0,t}, \mathbf{p}_{0,t}, \mathbf{p}_{0,t}), \end{aligned} \tag{11}$$

where $\mathbf{p}_{0,t}$ is the coordinate of thumb of the t-th frame, and $\mathbf{p}_{I,t}, \mathbf{p}_{M,t}, \mathbf{p}_{R,t}$ and $\mathbf{p}_{L,t}$ are the coordinates of index fingertip, middle fingertip, ring fingertip and little fingertip at frame t respectively. Finally, $\mathbf{v_t}$ of each frame is concatenated as the FRPV:

$$\mathbf{V_{FRPV}} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t, \ldots, \mathbf{v}_T). \tag{12}$$

3.5. HPEV-Net and HMM-Net

We learn the discriminative hand posture variations from HPEV using the 3D CNN based HPEV-Net. As shown in Figure 3, we first use a 3D convolution layer to extract the

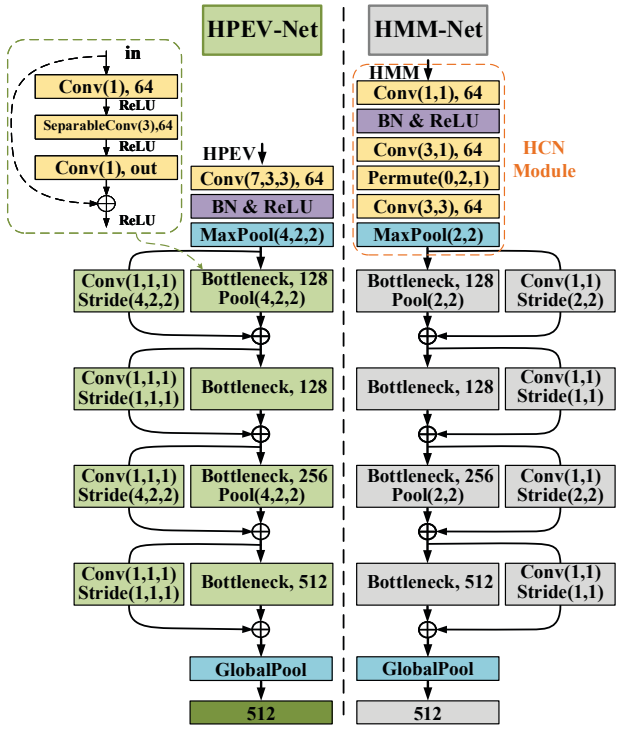

Figure 3. The network details of HPEV-Net and HMM-Net. The left is the HPEV-Net, and the right is the HMM-Net.

low-level features. Since there is a gap θ between adjacent skeletons in HPEV, the kernel size is set to $7 \times 3 \times 3$ to capture both the local spatial configurations and temporal information. Then, we stack four bottleneck modules [14] to learn high-level hand posture variation features gradually from the HPEV. Within the bottleneck module, the input features are turned into 64 channels using a $1 \times 1 \times 1$ convolution layer. In order to shrink the model, we use a depthwise separable convolution [5] with $3 \times 3 \times 3$ kernel size instead of common convolution in the bottleneck. The output of the depthwise separable convolution is followed by a $1 \times 1 \times 1$ convolution to match the output channels. For residual connection, the input features go through a $1 \times 1 \times 1$ convolution and the output is added to the output of the bottleneck. The output channels of the four bottleneck modules are 128, 128, 256 and 512. Three $4 \times 2 \times 2$ max pooling layers are used to reduce the size of output features in the first convolution layer and two bottleneck modules. Batch Normalization and ReLU are used after 3D convolution at each layer. The output features of the last bottleneck module are turned into a feature vector using global average pooling.

As shown in Figure 2, in order to remedy the resolution restriction of volume representation, the FRPV feature is concatenated to the hand posture variation feature vector as a supplementary cue for hand gesture recognition. We apply a fully connected layer with Batch Normalization and ReLU to the FRPV before concatenation, so as to guaranty that the two feature vectors are on the same order of magnitude. Finally, we append HPEV-Net with a fully connected layer to classify the hand gesture sequences.

We extract hand movement features from HMM using the CNN based HMM-Net. Our architecture is based on the co-occurrence module from Hierarchical Co-occurrence Network (HCN) [20]. We first use an HCN module to extract features. Similarly, four stacked bottleneck modules are used to learn high-level hand movement features and the output is turned into a feature vector using the global average pooling. Once again, we append the network with a fully connected layer. In the end, the scores provided by the HPEV-Net and HMM-Net are averaged to produce the final prediction.

4. Experiments and analysis

4.1. Datasets

We carry out experiments and perform comparison with related approaches on three public benchmarks: SHREC'17 Track, DHG-14/28 and FPHA dataset.

SHREC'17 Track. The SHREC'17 Track dataset [31] is a challenging hand gesture dataset, it contains 14 gestures performed by 28 individuals in two ways: using one finger and the whole hand. It comprises 2800 sequences, which are divided into 1960 sequences for training and 840 sequences for testing. Coordinates of 22 hand joints are provided for each skeleton.

DHG-14/28. The DHG-14/28 dataset [7] includes 14 gestures with 2800 sequences provided by 20 individuals. The DHG-14/28 dataset has the same hand joints and data collection method as the SHREC'17 Track dataset. We use the leave-one-subject-out experimental protocol for training and testing [7, 24, 4].

FPHA. The FPHA dataset [13] provides first-person dynamic hand action sequences performed by 6 actors. It comprises 1175 action samples, including 45 categories manipulating 26 different objects in 3 scenarios. 3D coordinates of 21 hand joints as the SHREC'17 Track dataset except for the palm joint are provided. We use a 1:1 setting with 600 action sequences for training and 575 for testing [13, 24].

4.2. Training details

All experiments are conducted using PyTorch on NVIDIA TITAN Xp. Adam is applied as the optimization strategy. Cross-entropy is selected as the loss function. As shown in Figure 2, the scores of two subnets are fused to produce the final prediction. This final prediction is used to compute the cross-entropy loss and the loss is back-propagated jointly for the two sub-networks. The batch size for training is 40. The initial learning rate is 3e-4 and the learning rate is reduced by a factor of 10 once learning stagnates. The training process is stopped when the learning

Table 1. Recognition accuracy (%) of our method for different input combinations on SHREC'17 Track dataset and FPHA dataset. 14G and 28G represent 14 and 28 gesture settings.

Method	SHREC		FPHA
	14G	28G	
HPEV	73.45	71.43	77.04
HMM	92.74	86.67	66.78
FRPV	62.86	58.81	66.43
HPEV+HMM	94.40	90.24	82.96
HPEV+HMM+FRPV	**94.88**	**92.26**	**90.96**

Table 2. Performance on fine and coarse category of SHREC'17 Track dataset with 14 gestures protocol.

Method	Fine	Coarse
HPEV	84.84	67.85
HMM	92.54	93.14
FRPV	83.75	52.58
HPEV+HMM	94.95	94.14
HPEV+HMM+FRPV	**95.31**	**94.67**

rate reaches 3e-8. The default settings of other parameters are $T = 60$, $\theta = 3$ and $R = 32$ as described in Section 4.3.

The parameters and FLOPs of the overall framework with default settings are 2.05M and 1.46G respectively. It achieves about 70 gestures per second for testing. This demonstrates that our method is efficient due to the specialized design of the framework. Hence, our method has great potential for real-world applications.

4.3. Ablation study

In this section, we explore the influence of different components and settings of our method. The default settings of parameters are $T = 60$, $\theta = 3$ and $R = 32$.

Different input combinations. To examine the influence of different input descriptor combinations (HPEV, HMM, FRPV), we conduct ablation experiments on both SHREC'17 Track and FPHA dataset. Results shown in Table 1 confirm that these three descriptors are critical for gesture recognition indeed. Combining all three descriptors achieves the best performance consistently. Note that HMM contributes more than HPEV on the SHREC'17 Track dataset, while the results are reversed on the FPHA dataset. The reason is that half of the gestures in the SHREC'17 Track dataset are about swiping, which have high correlations with hand movements. Moreover, using FRPV obtains about a 8% increase for accuracy on the FPHA dataset, since most gestures in the FPHA dataset involve subtle motion, *e.g.* writing, reading letters, cleaning glasses, and so on.

There are two groups in the SHREC'17 Track dataset,

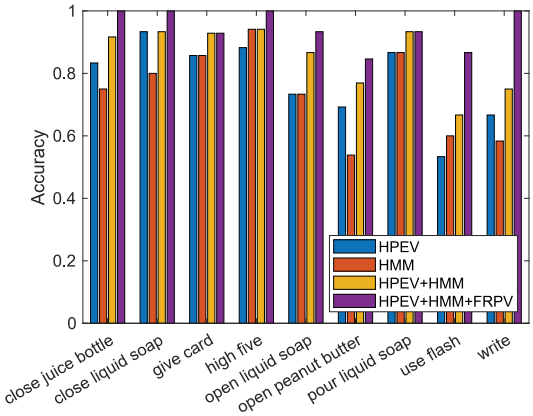

Figure 4. Comparison of recognition accuracy using different input combinations for some gestures on FPHA dataset.

Table 3. Recognition accuracy (%) of our method for different HPEV structures on SHREC'17 Track dataset and FPHA dataset. 14G and 28G represent 14 and 28 gesture settings.

Method	SHREC		FPHA
	14G	28G	
HPEM+HMM+FRPV	92.85	90.35	85.74
HPEV+HMM+FRPV	**94.88**	**92.26**	**90.96**

fine and coarse gestures. In order to explore the performance of different input combinations on the fine and coarse gestures, the recognition accuracies of these two categories on the SHREC'17 Track dataset with 14 gestures protocol are listed in Table 2. Fine gestures always involve hand posture changes, while coarse gestures involve hand movements. Therefore, it is reasonable that both HPEV and FRPV perform much better on fine gestures, while HMM performs better on coarse gestures.

The recognition accuracy of some gesture classes on the FPHA dataset is provided in Figure 4. The HPEV+HMM input combination usually outperforms both the HPEV and the HMM, showing the complementary property of HPEV and HMM. The FRPV is beneficial to subtle gestures. Especially for the gesture "write", the FRPV achieves a performance boost over 20%. The results demonstrate the effectiveness of our approach for hand gesture modeling.

Influence of HPEV structure. To investigate the influence of the HPEV structure, we use HMM like structure to model the hand posture variations as Hand Posture Evolution Map (HPEM). The performance of HPEV and HPEM is given in Table 3. The results for HPEM on SHREC'17 Track and FPHA dataset are about 2%, 2% and 5% decrease compared with HPEV. This shows the effectiveness of the HPEV structure for modeling hand posture variations, especially for the FPHA dataset which contains numerous gestures involving hand posture changes.

Table 4. Recognition accuracy (%) of our method for different gaps θ on SHREC'17 Track dataset with 14 gestures protocol.

Gap (θ)	HPEV	HPEV+HMM+FRPV
0	62.26	93.10
1	70.90	93.93
2	72.74	93.93
3	**73.45**	**94.88**
5	72.86	94.76

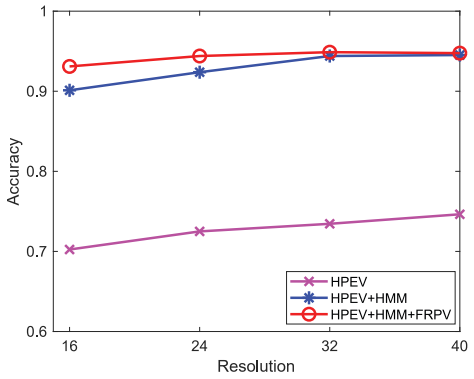

Figure 5. Recognition accuracy of our method for different resolution R on SHREC'17 Track dataset with 14 gestures protocol.

Different gaps. To explore the influence of different gaps, we conduct the ablation study on the SHREC'17 Track dataset with 14 gestures protocol. In this experiment, the gap θ between adjacent skeletons varies from 0 to 5, other settings are kept unchanged. Table 4 shows the performance of our method with different gap settings. Note that the accuracy decreases rapidly for $\theta = 0$ when using only the HPEV feature. This shows the gap is important for hand posture evolution volume modeling. Results suggest that using gap $\theta = 3$ is sufficient to obtain good performance.

Different resolution. To investigate the influence of different resolution when building HPEV, we conducted ablation experiments on the SHREC'17 Track dataset with 14 gestures protocol. Results with $R = 16, 24, 32, 40$ are shown in Figure 5. Performance gets worse with lower resolution. Results suggest that using resolution $R = 32$ is sufficient to achieve good performance. Comparing the two curves corresponding to HPEV+HMM and HPEV+HMM+FRPV, it is obvious that when the resolution is increased, the FRPV helps gain less recognition accuracy.

4.4. Comparison with the state-of-the-art

We perform the experimental comparison with several state-of-the-art approaches on the DHG-14/28, SHREC'17 Track and FPHA dataset respectively. The results are shown and discussed as follows.

SHREC'17 Track dataset. The state-of-the-art methods we used for comparison on SHREC'17 Track dataset are divided into five categories: 1) **Hand-crafted methods**: HON4D [28], Devanne *et al*. [8], Ohn-Bar *et al*. [26], SoCJ+Direction+Rotation [6], SoCJ+HoHD+HoWR [7], Caputo *et al*. [2] and Boulahia *et al*. [1]; 2) **CNN-based methods**: SEM-MEM+WAL [21], Res-TCN [15] and STA-Res-TCN [15]; 3) **3D-CNN-based method**: Two-stream 3D CNN [32]; 4) **Graph-based methods**: ST-GCN [36] and DG-STA [4]; 5) **Manifold-learning-based method**: ST-TS-HGR-NET [24].

Table 5. Recognition accuracy and comparison with the state-of-the-art approaches on SHREC'17 Track dataset with 1960 sequences for training and 840 sequences for testing. 14G and 28G represent 14 and 28 gesture settings.

Method	Accuracy (%)	
	14G	28G
HON4D [28]	78.50	74.00
Devanne *et al*. [8]	79.40	62.00
Ohn-Bar *et al*. [26]	83.90	76.50
SoCJ+Direction+Rotation [6]	86.90	84.20
SoCJ+HoHD+HoWR [7]	88.20	81.90
Caputo *et al*. [2]	89.50	-
Boulahia *et al*. [1]	90.50	80.50
Two-stream 3D CNN [32]	83.45	77.43
SEM-MEM+WAL [21]	90.83	85.95
Res-TCN [15]	91.10	87.30
STA-Res-TCN [15]	93.60	90.70
ST-GCN [36]	92.70	87.70
ST-TS-HGR-NET [24]	94.29	89.40
DG-STA [4]	94.40	90.70
Ours	**94.88**	**92.26**

Table 5 lists the recognition accuracy of all methods. Two-stream 3D CNN [32] and SEM-MEM+WAL [21] are closely related to our method. We implemented these two approaches and applied them on SHREC'17 Track dataset. The results of other methods are collected from papers [4, 24]. Our approach achieves state-of-the-art performance under both the 14-gesture and 28-gesture settings. In particular, our method obtains 94.88% on the 14-gesture protocol and 92.26% on the 28-gesture protocol. It outperforms the most recent work DG-STA [4] by 0.48% and 1.5% for experiments with 14 and 28 gestures respectively.

DHG-14/28 dataset. In addition to the methods listed in Table 5, we also compare with: a RNN-based method [3], a CNN-based method [35] and a LSTM-based method [25] on DHG-14/28 dataset. Table 6 shows that our method achieves state-of-the-art performance under both the 14-gesture and 28-gesture settings on the DHG-14/28 dataset. We collect the results of other baseline methods from papers [4, 24]. Although the DHG-14/28 dataset has the same hand gestures with the SHREC'17 Track dataset, it is more

Table 6. Recognition accuracy and comparison with the state-of-the-art methods on DHG-14/28 dataset using the leave-one-subject-out experimental protocol. 14G and 28G represent 14 and 28 gesture settings.

Method	Accuracy (%)	
	14G	28G
SoCJ+HoHD+HoWR [7]	83.10	80.00
Chen *et al*. [3]	84.70	80.30
CNN+LSTM [25]	85.60	81.10
Weng *et al*. [35]	85.80	80.40
Res-TCN [15]	86.90	83.60
STA-Res-TCN [15]	89.20	85.00
ST-GCN [36]	91.20	87.10
ST-TS-HGR-NET [24]	87.30	83.40
DG-STA [4]	91.90	88.00
Ours	**92.54**	**88.86**

challenging due to the leave-one-subject-out experimental protocol. As shown in Table 6, our method obtains 92.54% on 14-gesture protocol and 88.86% on 28-gesture protocol. It outperforms the most recent work DG-STA [4] by 0.64 and 0.86 percent point for experiments with 14 and 28 gestures respectively. Note that the good performance is more notable with 28 gestures setting than that with 14 gestures setting. We argue that our HPEV for hand skeleton encodes the spatial configurations of the hand effectively. Thus, the network can distinguish the gestures performed using one finger and the whole hand efficiently.

FPHA dataset. FPHA dataset is a new and challenging dataset for studying first-person dynamic hand actions interacting with 3D objects. Compared with the SHREC'17 Track dataset and the DHG-14/28 dataset, it includes more gesture categories, *i.e*. 45 daily hand action classes. Due to the obvious inter-subject and intra-subject variations on style, speed, scale, and viewpoint, to recognize the hand action sequences is a challenge.

Table 7 shows the recognition accuracy of our approach and recent methods on FPHA dataset. We quote the results of compared methods from paper [24]. Our method achieves competitive performance. Note that ST-TS-HGR-NET [24] outperforms our method. However, on the other hand, our approach is superior to ST-TS-HGR-NET both on the SHREC'17 Track dataset and DHG-14/28 dataset. Particularly for the DHG-14/28 dataset, our method outperforms ST-TS-HGR-NET by 5.24 and 5.46 percent points for 14 gestures and 28 gestures protocol. ST-TS-HGR-NET is based on manifold learning with SVM classifier. It has better generalization ability on the FPHA dataset which is a small dataset with 600 samples for training. For the DHG-14/28 dataset with about 2660 samples for training, it lacks learning capacity for the larger dataset. However, our method is based on deep networks showing powerful learning ability for large datasets.

Table 7. Performance comparison with the state-of-the-art methods on FPHA dataset. C, D, P represent color, depth and pose.

Method	C	D	P	Acc. (%)
Two stream-color [11]	✓	✗	✗	61.56
Two stream-flow [11]	✓	✗	✗	69.91
Two stream-all [11]	✓	✗	✗	75.30
HOG^2-depth [27]	✗	✓	✗	59.83
HOG^2-depth+pose [27]	✗	✓	✓	66.78
HON4D [28]	✗	✓	✗	70.61
Novel View [29]	✗	✓	✗	69.21
JOULE-color [16]	✓	✗	✗	66.78
JOULE-depth [16]	✗	✓	✗	60.17
JOULE-pose [16]	✗	✗	✓	74.60
JOULE-all [16]	✓	✓	✓	78.78
1-layer LSTM [39]	✗	✗	✓	78.73
2-layer LSTM [39]	✗	✗	✓	80.14
Moving Pose [37]	✗	✗	✓	56.34
Lie Group [33]	✗	✗	✓	82.69
HBRNN [10]	✗	✗	✓	77.40
Gram Matrix [38]	✗	✗	✓	85.39
TF [12]	✗	✗	✓	80.69
Huang *et al*. [17]	✗	✗	✓	84.35
Huang *et al*. [18]	✗	✗	✓	77.57
ST-TS-HGR-NET [24]	✗	✗	✓	**93.22**
Ours	✗	✗	✓	90.96

5. Conclusion

We propose a new method for hand gesture modeling. The gesture is decomposed into hand posture variations and hand movements, which are encoded into the HPEV and the HMM respectively. We introduce a framework including the HPEV-Net and HMM-Net to learn these two features for gesture recognition. Due to the specialized design of the framework, our method has great potential for real-world applications. Extensive experiments demonstrate that our method is competitive or superior to related work.

Acknowledgement

The authors thank anonymous reviewers very much for their valuable comments. This research was supported by the Major Project for New Generation of AI under Grant No. 2018AAA0100400, the National Natural Science Foundation of China under Grants 91646207, 61976208, 61620106003, and 61773377, and the Beijing Natural Science Foundation under Grant 4162064.

References

[1] Said Yacine Boulahia, Eric Anquetil, Franck Multon, and Richard Kulpa. Dynamic hand gesture recognition based on 3d pattern assembled trajectories. In *IPTA*, pages 1–6. IEEE, 2017.

[2] Fabio M Caputo, Pietro Prebianca, Alessandro Carcangiu, Lucio D Spano, and Andrea Giachetti. Comparing 3d trajectories for simple mid-air gesture recognition. *Computers & Graphics*, 73:17–25, 2018.

[3] Xinghao Chen, Hengkai Guo, Guijin Wang, and Li Zhang. Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition. In *ICIP*, pages 2881–2885. IEEE, 2017.

[4] Yuxiao Chen, Long Zhao, Xi Peng, Jianbo Yuan, and Dimitris N Metaxas. Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. In *BMVC*, 2019.

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1251–1258, 2017.

[6] Quentin De Smedt. *Dynamic hand gesture recognition-From traditional handcrafted to recent deep learning approaches*. PhD thesis, 2017.

[7] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Skeleton-based dynamic hand gesture recognition. In *CVPRW*, pages 1–9, 2016.

[8] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE transactions on cybernetics*, 45(7):1340–1352, 2014.

[9] Guillaume Devineau, Fabien Moutarde, Wang Xi, and Jie Yang. Deep learning for hand gesture recognition on skeletal data. In *FG*, pages 106–113. IEEE, 2018.

[10] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, pages 1110–1118, 2015.

[11] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016.

[12] Guillermo Garcia-Hernando and Tae-Kyun Kim. Transition forests: Learning discriminative temporal transitions for action recognition and detection. In *CVPR*, pages 432–440, 2017.

[13] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *CVPR*, pages 409–419, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] Jingxuan Hou, Guijin Wang, Xinghao Chen, Jing-Hao Xue, Rui Zhu, and Huazhong Yang. Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition. In *ECCV*, pages 0–0, 2018.

[16] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *CVPR*, pages 5344–5352, 2015.

[17] Zhiwu Huang and Luc Van Gool. A riemannian network for spd matrix learning. In *AAAI*, 2017.

[18] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on grassmann manifolds. In *AAAI*, 2018.

[19] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *CVPR*, pages 3288–3297, 2017.

[20] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *IJCAI*, 2018.

[21] Hanying Liu, Juanhui Tu, Mengyuan Liu, and Runwei Ding. Learning explicit shape and motion evolution maps for skeleton-based human action recognition. *ICASSP*, pages 1333–1337, 2018.

[22] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *CVPR*, pages 1647–1656, 2017.

[23] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[24] Xuan Son Nguyen, Luc Brun, Olivier Lézoray, and Sébastien Bougleux. A neural network based on spd manifold learning for skeleton-based hand gesture recognition. In *CVPR*, pages 12036–12045, 2019.

[25] Juan C Nunez, Raul Cabido, Juan J Pantrigo, Antonio S Montemayor, and Jose F Velez. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*, 76:80–94, 2018.

[26] Eshed Ohn-Bar and Mohan Trivedi. Joint angles similarities and hog2 for action recognition. In *CVPRW*, pages 465–470, 2013.

[27] Eshed Ohn-Bar and Mohan Manubhai Trivedi. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE transactions on intelligent transportation systems*, 15(6):2368–2377, 2014.

[28] Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *CVPR*, pages 716–723, 2013.

[29] Hossein Rahmani and Ajmal Mian. 3d action recognition from novel viewpoints. In *CVPR*, pages 1506–1515, 2016.

[30] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016.

[31] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, Joris Guerry, Bertrand Le Saux, and David Filliat. Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset. In *3DOR*, 2017.

[32] Juanhui Tu, Mengyuan Liu, and Hanying Liu. Skeleton-based human action recognition using spatial temporal 3d convolutional neural networks. *ICME*, pages 1–6, 2018.

[33] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, pages 588–595, 2014.

[34] Hongsong Wang and Liang Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *CVPR*, pages 499–508, 2017.

[35] Junwu Weng, Mengyuan Liu, Xudong Jiang, and Junsong Yuan. Deformable pose traversal convolution for 3d action and gesture recognition. In *ECCV*, pages 136–152, 2018.

[36] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[37] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *ICCV*, pages 2752–2759, 2013.

[38] Xikang Zhang, Yin Wang, Mengran Gou, Mario Sznaier, and Octavia Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *CVPR*, pages 4498–4507, 2016.

[39] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, 2016.